

LOADSTAR LETTER

Number 32. Companion the LOADSTAR disk #142

Just when You Thought You Were "Compleatly Safe," LOADSTAR Launches Two New Software Titles

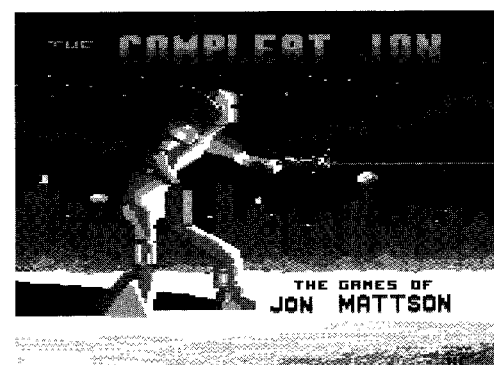
LOADSTAR proudly introduces two new titles in our *Compleat* series: *The Compleat Crossword* and a new game disk, *The Compleat Jon*. *The Compleat Crossword* is a major collection of 220 crossword puzzles found in *Puzzle Page* on LOADSTAR.



From a handy scrolling menu you can select any of the puzzles and then solve them on screen with Barbara Schulak's *Cruciverbalist* program. When done with the puzzle you have the option of "marking" the puzzle as one that's solved.

Compleat Crossword is a testament of digital cooperation, compiled by Bill Calvert, Presenter by Fender Tucker, and Puzzle Program by Barbara Schulak. The myriad of puzzles were crafted by Barbara Schulak, Fender Tucker, Craig Buchman, Linda Vano, Peter Rokitski and Knees Calhoon Music by Dave Marquis. You can even select which of the nine songs you'd like to hear while solving the puzzles.

The Compleat Jon, The Games of Jon Mattson, is a collection of 11 superb games previously published on LOADSTAR. During



LOADSTAR's "middle years" (1987-1990) one programmer stood out from the crowd because of the sophistication of his programs -- Jon Mattson. Practically every month he would send LOADSTAR his latest creation, which usually used the most modern techniques of game programming.

The whole gamut of gaming is covered here: artificial intelligence, role-playing, mazes, fantasy, science fiction, education and even non-violence (which was a radical concept in its time).

The thing I like best about the games is that each takes place in a consistent, realized world of its own. Read the docs, which are all by Jon himself, to get the background story behind each game. These are not silly little games made with character graphics; they are games with backgrounds and stories, and Jon tells these stories very well, indeed. By the time you finish the docs, you are *in* the created world, and once you run the program you *are* the hero, with lives depending on you and your cleverness and dexterity.

Things That Make You Go "Hmmm..."

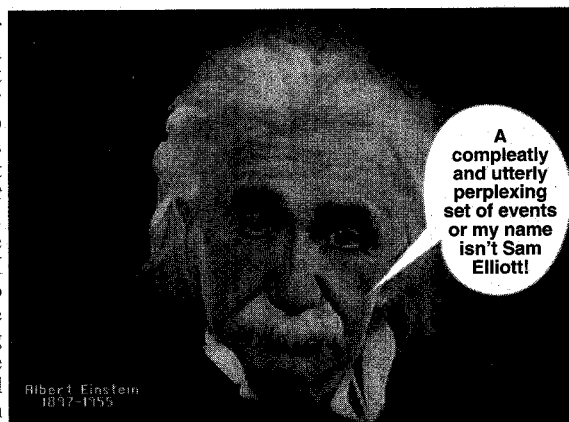
by Jeff Jones. While developing *The Compleat Crossword*, Fender stumbled upon a bug in my [Jeff] code that causes a crash whenever following 6502 code creates an integer variable, F%, with a value of 40.

```
;lsb in 253
;msb in 254
lda "F"
sta 251
lda "F"
sta 252
lda $7a
pha
lda $7b
pha
lda <251
sta $7a
lda >251
sta $7b
jsr $b08b
ldy #0
lda 254 ;high byte
sta ($47),y
iny
lda 253 ;low byte
sta ($47),y
pla
sta $7b
pla
sta $7a
rts
```



This code was handed down to me from C-64 programming legend, Rick Nash. I'll admit that I've mutated the code a bit. I use this code all the time to pass data back to BASIC from my machine language routines. I used this code in *Menu Toolbox*, which Fender used in *Compleat Crossword*. When Fender selected the 40th item on the crossword menu, the program crashed with a syntax error. I studied the code and found nothing that should be agitated by a value of 40. I plugged in a 40 and assembled the code by itself. *Crash!* What's more weird, *Toolbox 105* uses similar code with different memory locations to protect its variables. It *doesn't* crash on 40. I can't figure out why. I was able to fix the code by making the variable zero, and then manually plugging in the integer value, using the \$5f pointer to the variable data.

I found the pointer by locating a variable in memory, and then using Super Snapshot V5.5.2's machine language monitor to search for any zero page locations pointing to the variable. Though this bug is squashed, I can't help but wonder what could have caused such a thing to happen. Hmmm. Anyone have any ideas?



Questions: 1-318-221-8718

Xnet, The Newest Commodore BBS Network

If you are currently running an Image BBS and want to join a network then look no further! JOIN XNET!! Just leave feedback (not Email) on the LOADSTAR BBS at 1-318-425-4382, saying that you wish to join! We would love to have you in our network!! No harsh rules! Just a bunch of Sysops having FUN!

Jeff's Eight Inches

On the LOADSTAR LETTER #25, I slammed the STAR SJ-144 color printer. I got one letter in its defense last fall. For the way the writer used the printer (a cool color graphic covering 20% of the paper in the middle of text), it seemed to have decent output. It still doesn't stack up to modern color ink jet printers. The Star SJ-144 isn't history because Jeff doesn't like it. It's history because it doesn't stack up. And how dare they call it "laser quality." It's hard enough to keep a straight face when you call a serious ink jet printer "laser quality."

Frank Vanaman wrote me, informing me that the Casio digital camera's 320x200 resolution wasn't sufficient for stunning photos, and that it was unfair to compare it with the scan of Fender and Judy in the LOADSTAR T-shirt ad. He also informed me that the 16 million color photos can make a photo look real at first glance, but at 320x200 pixels, you can still see the lack of detail.

I agree. Also one thing some people don't understand about "16 million color pictures" is that a 320x200 image only has 64,000 pixels, and if each were a different color, it would only be a 64,000 dot picture. There are 16 million colors available in a 24-bit picture. I just left Kroger where I perused a Modern photography. They asked if digital photos would overtake 35mm. They did side-by-side comparisons of digital cameras, including the Casio QV-10, and all the digital cameras paled to old fashioned 35mm cameras. To add insult to injury, they took a picture with a lowly disposable camera, then scanned it, and came up with a better picture!

But there was a line of professional digital cameras that ranged from a few thousand to \$27,000, which took beautiful sharp photos at more than 1000x1000 pixels. I was impressed. Maybe in ten years, I can get a camera like that for a few hundred. Until then, digital artists with non-corporate budgets will continue to rely on "real" film cameras. I'm looking at a good focusable Polaroid camera for use with my scanner.

While editing a program this month, I came upon an interesting way to mix Commodore uppercase graphics with Upper/lowercase text on a Commodore printer. Open two channels to the printer. Open4,4,7:open5,4,0. Print#4 for text and print#5 for graphics. Why had I never thought of it?

I got a piece of Email, stating that LOADSTAR pushes CMD drives, but doesn't support them fully. My answer to that is that is simple: We support CMD drives in the best way possible -- by making our programs compatible, and making them run on any device number when possible. CMD designs their drives and RAM devices to be so transparent, it's difficult to think of a way to support them specifically except for partition and subdirectory surfing. I use all sorts of programs to navigate CMD devices. These programs weren't necessarily designed to be used on a CMD device.

If the program uses or creates files, it will either allow the sending of disk commands or long filenames. That's all I need for CMD support. It's one reason why I never ordered the HD version of TWS. My old version of The Write Stuff works fine on my RAMLink.

Again, LOADSTAR's main reason for pushing the FD-2000 is that one day there will be no 1541s in working order. That, and there will be no more new 5.25-inch disks. Can you imagine LOADSTAR falling because it's loyal readers had no way to RUN its software?

A Commie Uprising: Internet Service Provider Revolt

"Internet Access And You" By: X-> MIKE <-X @ DMB The basic unit of computing is the BIT... a state determined by an ON voltage and an OFF (or null) voltage. The basic unit of networking is to transmit one bit from computer A to computer B. 8 bits make up one byte, which is equivalent to one character/letter on your screen. The 'internet' is just a collection of many networked computers. 'Internet' connectivity is highly variable, from real-time supercomputer connections to even Coca-cola machines (yep!).

The WORLD WIDE WEB is NOT the 'Internet'. It simply uses the

Internet as a carrier. A good analogy is to say "my boat is the ocean". No, it is not! Your boat simply rides on the ocean.

USENET is NOT the 'Internet'. It also uses the Internet as a carrier.

The 'internet' is just a collection of many networked computers.

You DO NOT need 'THE WAVE' to access the 'Internet'. You MAY need 'THE WAVE' to access the GRAPHICAL portion of the WORLD WIDE WEB. You *certainly do not* need 'THE WAVE' to access the TEXTual portion of the WORLD WIDE WEB.

The WORLD WIDE WEB is just a *filesystem*! Any vt-100 capable term program should be sufficient for internet connectivity as vt-100 (and in reality, pure ascii) is the primary currency for information interchange in the known world. If your internet provider refuses to acknowledge this, find another provider.

Q From Caped Crusader: I don't see why we can't eventually hook up our Commodore boards with Telenet and folks could call a Commodore 8 bit board using the Internet. ----

A: Telenet... telNet was (is?) an old and hopefully-now-obsolete and useless network that charged you up the wahzoo for... err, wow...slow down Mike..

I have done what you propose. I 'telnetted' a DIAL-OUT and had it call a local BBS. To be more precise, Highlander BBS in Toronto (416). Also, some guy put his BBS on the net, in a matter of speaking. He has a pc running linux acting as a go-between for the commodore. The BBS is so-so and the connection is no different than the DIAL-OUT idea (in essence you use a more pro-actively connected machine to spoon feed the commodore). What I am waiting for is an actual TCP/IP stack for the Commodore itself, which would allow it what I consider 'true' internet connectivity. It's like the difference between running a basic program and claiming you program in machine language (essentially you do, since basic is written in ML :) and actually getting down and doing ML.

PS: DO NOT ASK ME FOR A DIAL-OUT. THESE THINGS ARE SO POPULAR AND GET ABUSED THEY RARELY LAST FOR MORE THAN A FEW WEEKS.

Q From Realm Master: You defined the internet as bunch of "networked computers". Well, the Commodore Image network is a bunch of networked computers so would one think that it would be okay to call our CommNet an Internet? By your definition, we are an internet....one may argue that the word "internet" might mean an international network. I guess that qualifies CommNet as an Internet if for no other reason than we have some network nodes in Canada. USA-Canada..international, right? -----

A: God, you said that with loads o sardony... but did you realize just how much truth you were spouting out? The only differences (from a nice 'big picture' view) between our networked C=s and the "internet" are basically that links between machines are usually real-time (well...tries to be) super-fast (compared to our stuff) almost always 100% automated, and the machines themselves are generally serious stuff that would set you back \$20 grand ez. But this is oh-so-general... Every time I connect to my IP server, in effect, my little 128 becomes part of the "internet" .. please notice the caveat about "true connectivity" in the reply to CC.

XmX

CMD Considered Including Swiftlink In Super CPU. Comments That Utility Cartridges May Work Only In 1 Mhz Mode.

From: CMD Sales <cmd.sales@the-spa.com>

To: H.Pieters@net.HCC.nl

Subject: Re: supercpu.128

Yes, SwiftLink will be compatible with the CPU. CMD is striving to maintain compatibility with as many cartridge port peripherals as possible including RAMLink, SwiftLink, SID Symphony and the EX3/2+1 expander. As for utility cartridges such as Super Snapshot, their compatibility may be limited to 1 Mhz modes because of their design and the fact that they take over the Kernal ROM when they are plugged in. And to your second suggestion, the answer is Yes. We did consider a built-in SwiftLink. However, it would have added too much to the units cost and would be a duplication for the many thousands of

ADC				
Add memory to Accumulator with carry				
Registers affected: NZCIDL				
Addressing mode	Assembly form	opcode	bytes	cycles
immediate	adc #oper	69	2	2
zero page	adc oper	65	2	3
zero page, x	adc oper,x	75	2	4
absolute	adc oper	6D	3	4
absolute, x	adc oper,x	7D	3	4*
absolute, y	adc oper,y	79	3	4*
(indirect, x)	adc (oper,x)	61	2	6
(indirect), y	adc (oper),y	71	2	5*

AND				
And memory to Accumulator				
Registers affected: NZCIDL				
Addressing mode	Assembly form	opcode	bytes	cycles
immediate	and #oper	29	2	2
zero page	and oper	25	2	3
zero page, x	and oper,x	35	2	4
absolute	and oper	2D	3	4
absolute, x	and oper,x	3D	3	4*
absolute, y	and oper,y	39	3	4*
(indirect, x)	and (oper,x)	21	2	6
(indirect), y	and (oper),y	31	2	5

ASL				
Shift left one bit (memory or accumulator)				
Registers affected: NZCIDL				
Addressing mode	Assembly form	opcode	bytes	cycles
accumulator	asl	0A	1	2
zero page	asl oper	06	2	5
zero page, x	asl oper,x	16	2	6
absolute	asl oper	0E	3	6
absolute, x	asl oper,x	1E	3	7

BCC				
Branch to address if carry flag is clear				
Registers affected: NZCIDL				
Addressing mode	Assembly form	opcode	bytes	cycles
relative	bcc oper	90	2	2*

BCS				
Branch to address if carry flag is set				
Registers affected: NZCIDL				
Addressing mode	Assembly form	opcode	bytes	cycles
relative	bcs oper	B0	2	2*

BEQ				
Branch to address if zero flag is clear				
Registers affected: NZCIDL				
Addressing mode	Assembly form	opcode	bytes	cycles
relative	beq oper	F0	2	2*

BIT				
Copy bit 7 of memory to N and bit 6 to V				
Registers affected: NZCIDL				
Addressing mode	Assembly form	opcode	bytes	cycles
zero page	bit oper	24	2	3
absolute	bit oper	2C	3	4

BMI				
Branch to address if negative flag is set				
Registers affected: NZCIDL				
Addressing mode	Assembly form	opcode	bytes	cycles
relative	bmi oper	30	2	2*

BNE				
Branch to address if zero flag not set				
Registers affected: NZCIDL				
Addressing mode	Assembly form	opcode	bytes	cycles
relative	bne oper	D0	2	2*

BPL				
Branch to address if negative flag not set				
Registers affected: NZCIDL				
Addressing mode	Assembly form	opcode	bytes	cycles
relative	bpl oper	10	2	2*

BRK				
Force an interrupt (break)				
Registers affected: NZCIDL				
Addressing mode	Assembly form	opcode	bytes	cycles
implied	brk	00	1	7

BVC				
Branch to address if overflow flag is clear				
Registers affected: NZCIDL				
Addressing mode	Assembly form	opcode	bytes	cycles
relative	bvc oper	50	2	2*

BVS				
Branch to address if overflow flag is set				
Registers affected: NZCIDL				
Addressing mode	Assembly form	opcode	bytes	cycles
relative	bvs oper	70	2	2*

CLC				
Manually clear the carry flag				
Registers affected: NZCIDL				
Addressing mode	Assembly form	opcode	bytes	cycles
implied	clc	18	1	2

CLD				
Clear decimal mode				
Registers affected: NZCIDL				
Addressing mode	Assembly form	opcode	bytes	cycles
implied	cld	D8	1	2

CLI				
Clear interrupt disable bit (enable interrupts)				
Registers affected: NZCIDL				
Addressing mode	Assembly form	opcode	bytes	cycles
implied	cli	58	1	2

CLV				
Clear overflow flag				
Registers affected: NZCIDL				
Addressing mode	Assembly form	opcode	bytes	cycles
implied	clv	B8	1	2

CMP				
Compare memory with accumulator				
Registers affected: NZCIDL				
Addressing mode	Assembly form	opcode	bytes	cycles
immediate	cmp #oper	C9	2	2
zero page	cmp oper	C5	2	3
zero page, x	cmp oper,x	D5	2	4
absolute	cmp oper	CD	3	4
absolute, x	cmp oper,x	DD	3	4*
absolute, y	cmp oper,y	D9	3	4*
(indirect, x)	cmp (oper,x)	C1	2	6
(indirect), y	cmp (oper),y	D1	2	5

CPX				
Compare memory with X register				
Registers affected: NZCIDL				
Addressing mode	Assembly form	opcode	bytes	cycles
immediate	cpx #oper	E0	2	2
zero page	cpx oper	E4	2	3
absolute	cpx oper	EC	3	4

CPY				
Compare memory with Y register				
Registers affected: NZCIDL				
Addressing mode	Assembly form	opcode	bytes	cycles
immediate	cpy #oper	C0	2	2
zero page	cpy oper	C4	2	3
absolute	cpy oper	CC	3	4

DEC				
Decrement memory in address by one				
Registers affected: NZCIDL				
Addressing mode	Assembly form	opcode	bytes	cycles
zero page	dec oper	C6	2	5
zero page, x	dec oper,x	D6	2	6
absolute	dec oper	CE	3	6
absolute, x	dec oper,x	DE	3	7

DEX				
Decrement X register by one				
Registers affected: NZCIDL				
Addressing mode	Assembly form	opcode	bytes	cycles
implied	dex	CA	1	2

DEY				
Decrement Y register by one				
Registers affected: NZCIDL				
Addressing mode	Assembly form	opcode	bytes	cycles
implied	dey	88	1	2

EOR				
Exclusive-Or W/accumulator (opposite bits on)				
Registers affected: NZCIDL				
Addressing mode	Assembly form	opcode	bytes	cycles
immediate	eor #oper	49	2	2
zero page	eor oper	45	2	3
zero page, x	eor oper,x	55	2	4
absolute	eor oper	4D	3	4
absolute, x	eor oper,x	5D	3	4*
absolute, y	eor oper,y	59	3	4*
(indirect, x)	eor (oper,x)	41	2	6
(indirect), y	eor (oper),y	51	2	5*

INC				
Increment value in address by one				
Registers affected: NZCIDL				
Addressing mode	Assembly form	opcode	bytes	cycles
zero page	inc oper	E6	2	5
zero page, x	inc oper,x	F6	2	6
absolute	inc oper	EE	3	6
absolute, x	inc oper,x	FE	3	7

INX				
Increment value in X register by one				
Registers affected: NZCIDL				
Addressing mode	Assembly form	opcode	bytes	cycles
implied	inx	E8	1	2

INY				
Increment value in Y register by one				
Registers affected: NZCIDL				
Addressing mode	Assembly form	opcode	bytes	cycles
implied	iny	C8	1	2

JMP				
Jump to code at specified location				
Registers affected: NZCIDL				
Addressing mode	Assembly form	opcode	bytes	cycles
absolute	jmp oper	4C	3	3
(indirect)	jmp (oper)	6C	3	5

JSR				
Jump to subroutine. Save return address				
Registers affected: NZCIDL				
Addressing mode	Assembly form	opcode	bytes	cycles
absolute	jsr oper	20	3	6

Introduction to 6510 Machine Language

by Jeffrey L. Jones. I originally wrote this article way back when I was new to assembler. Since I was closer to the moment of "eureka" then, perhaps this article is the best way to explain the mysterious commands on these two pages.

The number one reason for writing programs in machine language is speed. A program written in machine language, even if sloppy code is used, can be thousands of times faster than even a compiled BASIC program. But most people, myself included, shudder at the thought of learning how to program in machine language. After all, BASIC was tough, and now that you've mastered it, you have no desire to master another language. Well, machine language is a lot easier to learn than BASIC, especially when you already know BASIC.

Basically you're dealing with memory locations. Everything in a machine language program is concerned with "What's in this location?", "Let's move that to another location", "Let's increment", "Let's decrement..."

You have three registers, A, X, and Y. Most every operation you do will be through the ACCUMULATOR or A register. This register in your CPU is the ONLY place where mathematics can occur. It also offers the most versatile forms of ADDRESSING. More on addressing soon. Conceivably entire programs could be written without accessing the ACCUMULATOR but I think it would cause high blood pressure problems.

Let's look at a simple, unelegant machine language program that moves lines 2 and 3 to lines 0 and 1 on your screen. All numbers used are in decimal unless a dollar sign precedes them. Most machine language monitors would force you to use HEX, and even if they allowed you to enter decimal with a + prefix would replace the decimal with the proper hex conversion when you hit RETURN. Assemblers allow you to type in the following program as is, without the need of the line numbers. Through the magic of the fact that this ain't a real disassembly listing, you get it both ways.

```
C000: LDY #0
C002: LDA 1104,Y
C005: STA 1024,Y
C008: INY
C009: CPY #80
C00B: BNE $C002
C00D: RTS
```

To write this program, you would need an assembler or a machine language monitor. You would also have to adhere to your monitor's protocol for entering the code. To RUN this program you would SYS 49152 (\$C000) though it's completely relocatable since there are no JMPs or JSRs inside the code.

Here's a BASIC equivalent:

```
FOR Y=0 TO 79: A=PEEK(1104+Y): POKE 1024+Y, A: NEXT
```

This program will PEEK the screen, starting at location 1104 (line 2) and then POKE it all, one by one, to line 0. The whole process takes 1.288 milliseconds to accomplish, while in BASIC it would take about two thousand times longer to execute -- literally.

The program starts out by LOADING the Y REGISTER with a 0 (in case it was anything else). Now we get to INDEXING: Note the command:

```
LDA 1102,Y is the same as:
A = PEEK(1102+Y)
```

This indexing is called ABSOLUTE, Y. It means you want to PEEK or POKE to Memory Location + Y

Y can be 0-255. So you can start with a base address and index to any of 256 addresses.

STA 1024,Y is the same as: POKE 1024+Y, A

Again we're using ABSOLUTE,Y indexing, this time to store the data in a different location. We're MOVING 80 bytes to a location 80 bytes lower, 1024 or the start of the default screen.

Next we have the commands:

```
C008: INY
C009: CPY #80
C00B: BNE $C002
```

These three commands are the equivalent to NEXT in BASIC. INY increments the Y register. If Y is incremented past 255, it rolls over to zero. CPY compares Y to the number 80 or the limit-1 of our loop.

BNE \$C002 is coupled with the previous command. If the comparison of Y to #80 hasn't occurred then BNE (BRANCH IF NOT EQUAL) to \$C002, which is the start of the loop. The BASIC command, NEXT does the same thing: Increments a variable, checks the variable against a limit, then branches to the beginning of the loop if the limit hasn't been met.

Finally we come to RTS, which is the same as the BASIC RETURN. RTS is short for RETURN FROM SUBROUTINE. Since you will more than likely SYS to this program from the immediate mode, RTS will RETURN to the immediate mode; the same as if you GOSUB to a BASIC subroutine from the immediate mode.

Besides vastly greater speed and more compact code, machine language offers more logical operators than BASIC. With BASIC you can basically branch if:

```
equal: IF A=X THEN...      not equal: IF A<>X THEN...
less or equal: IF A<=THEN... less than: IF A<X THEN
greater or equal: IF A>=THEN... greater than: IF A>X THEN...
if not 0 (true): IF A THEN...
```

*Add one cycle if page boundary is crossed. For branching, add 2. Each cycle is roughly 1 millionth of a second. In order to use these commands, you need a machine language monitor or preferably an assembler. Both of these are provided in The COMPLETE PROGRAMMER.

if equal.....	BEO	if zero.....	BEO
if not equal.....	BNE	if not zero (true).....	BNE
if less or equal.....	BCC	if less than.....	BCC
if greater than or equal.....	BCS	if greater than.....	BCS
if negative.....	BMI	if greater than 127.....	BMI
if positive.....	BPL	if less than 128.....	BPL
if overflow set.....	BVS	if bit 6 set (64).....	BVS
if on overflow clear.....	BVC	if bit 6 not set.....	BVC

Negative Number are all numbers greater than 127. 0 - 127 Are Positive. 128 - 255 = Negative. So as soon as you load that accumulator (or any other register with any number, the N (negative) flag is set for you. Coupled with the N flag are two commands:

Taking the previous chart into account, we can write our little line mover in a number of different ways, not to exclude counting backwards, which is my favorite way of looping.

```
C000: LDY #79
C002: LDA 1104,Y
C005: STA 1024,Y
C008: DEY
C009: CPY #255
C00B: BNE $C002
C00D: RTS
```

$$128 = \begin{matrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \wedge & \wedge & \wedge & \wedge & \wedge & \wedge & \wedge & \wedge \\ 128 & 64 & 32 & 16 & 8 & 4 & 2 & 1 \\ N & & & & & & & \end{matrix}$$

As soon as you load the accumulator with 128, flags go off:

1. Not equal to zero Z flag
2. A "negative" number: N flag

```
C000: LDY #79
C002: LDA 1104,Y
C005: STA 1024,Y
C008: DEY; affects N flag
C009: BPL $C002
C00B: RTS
```

Note that you can't use loops like this to move more than 129 bytes. This is because the loop would fall through on the first pass if Y were loaded with 129 or greater. This is because the N flag would be set after DEY since 128 is negative!

COMPLEAT PROGRAMMER

Programming BASICS
None at All
Applied BASIC
More Than
BASIC Utilities
Eight And Seven
Setting Boards
More Than

WHAT WE GOT NEED TO KNOW

Drive:11 Press H for Help Disk H 1

The preceding article was only 6000 bytes out of the **megabytes** of compressed knowledge found in **LOADSTAR's Compleat Programmer disks**. 8 3.5-inch disks or 2 3.5-inch disks. \$20. See page 2 for details. Call 1-800-594-3370 to order.

Load Accumulator				
Registers affected: NZCIV				
Addressing mode	Assembly form	opcode	bytes	cycles
immediate	lda #oper	A9	2	2
zero page	lda oper	A5	2	3
zero page, x	lda oper,x	B5	2	4
absolute	lda oper	AD	3	4
absolute, x	lda oper,x	BD	3	4*
absolute, y	lda oper,y	B9	3	4*
(indirect, x)	lda (oper,x)	A1	2	6
(indirect, y)	lda (oper,y)	B1	2	5*

LDX				
Load X register				
Registers affected: NZC/DV				
Addressing mode	Assembly form	opcode	bytes	cycles
immediate	ldx #oper	A2	2	2
zero page	ldx oper	A6	2	3
zero page, y	ldx oper,y	B6	2	4
absolute	ldx oper	AE	3	4
absolute, y	ldx oper,y	BE	3	4*

LDY				
Load Y register				
Registers affected: NZC/DV				
Addressing mode	Assembly form	opcode	bytes	cycles
immediate	ldy #oper	A0	2	2
zero page	ldy oper	A4	2	3
zero page, x	ldy oper,x	B4	2	4
absolute	ldy oper	AC	3	4
absolute, x	ldy oper,x	BC	3	4*

Halves value		LSR 0-76543210 - C
Shift right one bit (memory or accumulator)		
Registers affected: NZC1DV		
Addressing mode	Assembly form	opcode bytes cycles
accumulator	lsr a	4A 1 2
zero page	lsr oper	46 2 5
zero page, x	lsr oper,x	56 2 6
absolute	lsr oper	4E 3 6
absolute, x	lsr oper,x	5E 3 7

NOP					
No Operation					
Registers affected: NZCIV					
Addressing mode	Assembly form	opcode bytes	cycles		
implied	non	EA	1	2	

implied	nop	EA	1	2
ORA				
OR Accumulator with memory				
Registers affected: NZCIVD				
Addressing mode	Assembly form	opcode	bytes	cycles
immediate	ora #oper	09	2	2
zero page	ora oper	05	2	3
zero page, x	ora oper,x	15	2	4
absolute	ora oper	0D	3	4
absolute, x	ora oper,x	1D	3	4*
absolute, y	ora oper,y	19	3	4*
(indirect, x)	ora (oper,x)	01	2	6
(indirect), y	ora (oper,y)	11	2	5

PHA
Push (store) Accumulator on Stack
 Registers affected: **N Z C D V**
 Addressing mode Assembly form opcode bytes cycles
 implied pha 48 1 3

implied	php	48	1	3
PHP				
Push processor status on Stack				
Registers affected: NZCIDV				
Addressing mode	Assembly form	opcode	bytes	cycles
implied	php	08	1	3

implied	plp	08	1	3
PLA				
Pull Accumulator From Stack				
Registers affected: NZCIV				
Addressing mode Assembly form opcode bytes cycles				
implied	plp	68	1	4

implied	pla	68	1	4
PLP				
Pull processor status from stack				
Registers affected: NZCIV				
Addressing mode	Assembly form	opcode	bytes	cycles
i X		80	1	4

implied plp 28 1 4

ROL **[76543210-C-]**

Rotate Left one bit (memory or accumulator)

Registers affected: **NZCIV**

Addressing mode Assembly form opcode bytes cycles

accumulator	rol	2A	1	2
zero page	rol oper	26	2	5
zero page, x	rol oper,x	36	2	6
absolute	rol oper	2E	3	6
absolute, x	rol oper,x	3E	3	7

ROR C-76543210

Rotate right one bit (memory or accumulator)

Registers affected: NZC1DV

Addressing mode Assembly form opcode bytes cycles

accumulator	ror	6A	1	2
zero page	ror oper	66	2	5
zero page, x	ror oper,x	76	2	6
absolute	ror oper	6E	3	6
absolute, x	ror oper,x	7E	3	7

RTI				
Return from interrupt				
Registers affected: NZCIV				
Addressing mode	Assembly form	opCode	bytes	cycles
implied	rti	40	1	6

RTS				
Return from subroutine				
Registers affected: NZC1DV				
Addressing mode	Assembly form	opcode	bytes	cycles
implied	rts	60	1	6

SBC				
Registers affected: NZCIV				
Addressing mode	Assembly form	opcode	bytes	cycles
immediate	sbc #oper	E9	2	2
zero page	sbc oper	E5	2	3
zero page, x	sbc oper,x	F5	2	4
absolute	sbc oper	ED	3	4
absolute, x	sbc oper,x	FD	3	4*
absolute, y	sbc oper,y	F9	3	4*
(indirect, x)	sbc (oper,x)	E1	2	6
(indirect), y	sbc (oper),y	F1	2	5*

SEC
Set Carry flag
Registers affected: **NZCIV**
Addressing mode Assembly form opcode bytes cycles
implied sec 38 1 2

implied	sed	F8	1	2
SED				
Set decimal mode				
Registers affected: NZCIDV				
Addressing mode	Assembly form	opcode	bytes	cycles
implied	sed	F8	1	2

implied	sei	78	1	2
SEI				
Set interrupt disable status				
Registers affected: NZC-IDV				
Addressing mode	Assembly form	opcode	bytes	cycles
implied	sei	78	1	2

implied	sel	/8	1	2
STA				
Store accumulator in memory				
Registers affected: NZC10V				
Addressing mode	Assembly form	opcode	bytes	cycles
zero page	sta oper	85	2	3
zero page, x	sta oper,x	95	2	4
absolute	sta oper	8D	3	4
absolute, x	sta oper,x	9D	3	5
absolute, y	sta oper,y	99	3	5
(indirect, x)	sta (oper,x)	81	2	6
(indirect), y	sta (oper),y	91	2	6

STX				
Store X register in memory				
Registers affected: NZCIV				
Addressing mode	Assembly form	opcode	bytes	cycles
zero page	stx oper	86	2	3
zero page, y	stx oper,y	96	2	4
absolute	stx oper	8E	3	4

Store Y register in memory				
Registers affected: NZCVID				
Addressing mode	Assembly form	opcode	bytes	cycles
zero page	sty oper	84	2	3
zero page, x	sty oper,x	94	2	4
absolute	stv oper	8C	3	4

TAX
Transfer Accumulator to X register
Registers affected: **N Z C D V**

Addressing mode	Assembly form	opcode	bytes	cycles
implied	tax	AA	1	2

implied	tax	AA	1	2
TAY				
Transfer Accumulator to Y register				
Registers affected: NZC1DV				
Addressing mode Assembly form opcode bytes cycles				
implied	tax	A8	1	2

TSX
Transfer Stack pointer to X register
 Registers affected: **NZC/DV**
 Addressing mode: **Assembly form: opcode bytes: cycles:**

TXA
Transfer X register to Accumulator
Registers affected: NZCIV
Addressing mode: Assembly form: opcode bytes: cycles:

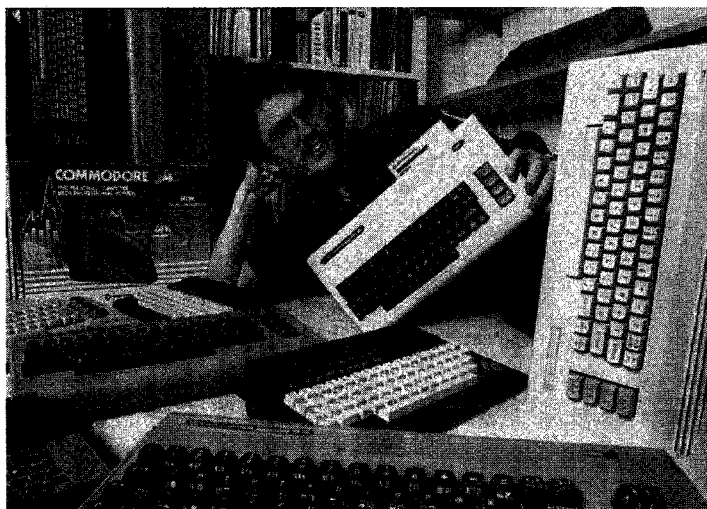
implied	txa	8A	1	2
TXS				
Transfer X register to Stack pointer				
Registers affected: N Z C I D V				
Addressing mode Assembly form opcode bytes cycles				
indirect TXS 01 1 6				

implied	txs	9A	1	2
TYA				
Transfer Y register to Accumulator				
Registers affected: NZCIV				
Addressing mode Assembly form opcode bytes cycles				

*Add one cycle if page boundary is crossed. For branching, add 2. Each cycle is roughly 1 millionth of a second. In order to use these commands, you need a machine language monitor or preferably an assembler. Both of these are provided in **The COMEALAT PROGRAMMER**.

people who already own SwiftLink. So, we won't be including it. Thanks for your interest and I hope that we can look forward to delivering a Super CPU to you as soon as they are shipping.

CMD Sales email: cmd.sales@the-spa.com
<http://www.the-spa.com/cmd/>



You, Your Commodore Computer, and the Internet, Part 5: Internet Electronic Mail

by Jim Brain. At the end of last class period, we were just delving into the simplest of all Internet services: electronic mail. Let us continue. Many of you have sent or received electronic mail, either via bulletin board systems, commercial services, or other means. Electronic mail is

simply a computerized version of regular paper mail. Every mail message has the following basic parts: Sender Address, Recipient Address, Message Body.

Like regular mail, Internet electronic mail (called e-mail or email) does not contain information on how to get the message from the sender to the recipient. Also, like its more traditional counterpart, email can be addressed in a variety of ways. However, it has a number of advantages over regular mail:

There is usually no charge to send or receive email. (The exception is CompuServe & MCI, which charge per kilobyte of mail read over a certain amount. The charge is trivial, but present.) Distance between source and destination is not a factor. (Even for CIS, the charge per kilobyte is the same no matter where the message came from.) Email is almost always faster. Most email arrives in 10 minutes or less, with 1 day being the maximum delivery time for almost all messages.

It's easy to send binary files and other non textual material in email messages. Sending to multiple people does not require copying of message for each individual. The paper saved is arguably more environmentally friendly.

Obviously, a few disadvantages include less personality (no special correspondence paper or fancy letterheads), computer/modem requirements, and less privacy. The last is important, as peeking at regular mail is considered a felony in many countries, but there is no such law in place for email. In fact, most email messages, if they must pass through a machine en route to their destination, are saved in areas that are easily accessible by those on that machine. Still, the sheer amount of email somewhat guarantees some element of privacy. However, if one is concerned about privacy, messages can be ciphered (just like playing detective when you were young).

Still, even with those drawbacks, Internet email is one of the most often used services of the Internet. In fact, almost all other Internet services can be accessed via email. An important advantage of email, of the Internet in general, is its anonymity. That is, if you don't tell someone your email was sent with a Commodore computer, they'll never know. So, let's send some email.

To send a piece of email, you must know the address of the person

SUBSCRIBE OR RENEW NOW AND WIN BIG!

CPU

the Commodore 128/64 Power User Newsletter presents its first ever subscription contest.

Ushering the Commodore User Into The 21st Century

GRAND PRIZE

3 SECOND PLACE PRIZES

11 THIRD PLACE PRIZES

CMD SUPER 64 CPU

CMD SMARTTRACK Trackball

GEOS ScanART vol. 1 disk

CPU

**P.O. Box 1817
Shelton, WA
98584**

The *Commodore 128/64 Power User* newsletter (CPU) is currently a quarterly publication aimed at providing timely and up-to-date information to any level of Commodore user. We bring to you informative and interesting articles on how to get the most out of your machine and software. Software and hardware reviews from a user's perspective. Find out about rare and powerful software that you probably didn't know existed. And of course what would a publication be without coverage of the Internet and telecommunications? The *Cyberspace Cowboy* (R.J. Smulkowski) covers this area in every issue! News, Reviews, Power using articles, Hints, Tips and much more in our ever-expanding coverage of the Commodore 8-bit scene! And now if you've been hesitant in the past to try CPU, you have the chance to get even more for your money: Subscribe (or renew your subscription) by July 1, 1996 and be automatically entered in our Fabulous Subscription Sweepstakes, you could win a **CMD Super64 CPU**, **CMD SmartTrack Trackball**, or a **GEOS ScanART** graphics collection disk from GGL, Inc. Publications! Winners' names will be announced in the *Loadstar Letter*, the *Summer 1996 CPU*, and *Commodore World* magazine. The grand and second place prizes will be mailed directly from CMD to the winners. **Please include a daytime phone number with your subscription order. Winners will be notified by mail if no phone number is given. Prizes will be shipped the second week of July 1996. Send Check or M.O. made payable to: CPU, P.O. Box 1817, Shelton, WA 98584. All funds must be in US Dollars. For more info, call our BBS and log on as GUEST, password CPU at 1-360-426-2406 or by voice ask for Tom, 1-360-426-0750. CPU reserves the right to substitute prizes if lack of availability warrants.**

6 Issue

Subscription:

United States \$15
 Can. and Mex. \$21
 Other Foreign \$27

Back Issues:

Back issues are available for \$3.50 each. This price includes shipping to U.S., Canada and Mexico. Foreign addresses add \$2.00 per issue.

you are sending the message to. Just like its traditional counterpart, an email address details exactly who the intended recipient is. For example:

Traditional Internet Description

Jim BrainbrainUsername 602 N. Lemenmailmachine Fentonmsendomain MI 48430comtop-level domain.

The addresses are organized like this:
TraditionalInternet

Jim Brainbrain@mail.msen.com 602 N. Lemen Fenton, MI 48430

Just as some people have addresses that omit the street address part, some Internet email addresses have no machine part.

Because so many people ask, there is NO directory of all email addresses. There are indeed catalogs of addresses available, but they are incomplete at best. At some point, we will such an exhaustive list, but not today.

The above addressing scheme is called the "Internet" style, and is most common. However, some systems still using UUCP (Unix to Unix Copy Program) instead of TCP/IP to transfer mail might use "bang" addresses. The above address in "bang" format would be:

mail.msen.com!brain

Speaking of "bang", Internet users have a shorthand way of pronouncing some symbols. These include:

!bang *star or splat .dot @at #hash

So, the address "brain@mail.msen.com" is pronounced: brain at mail dot msen dot com. Usually, a person must give you his or her email address via some other medium. However, for large commercial services, you can infer the Internet email address if you know the username:

GEnie JIMjim@genie.com DELPHIJIMjim@delphi.com
Compuserve 12345,67812345.678@compuserve.com (note ',' to '.' change) AOLjim01jim01@aol.com Prodigyjimjim@prodigy.com

Note that Internet email addresses are *not* case sensitive. jim@mail.msen.com, Jim@mail.msen.com, and JIM@MAIL.msen.com are all the same.

Now, you'll need a mail client program to send mail. For shell users, PINE is a good choice. Here is the opening screen:

PINE 3.91 MAIN MENU Folder: (CLOSED) 0 Messages

- ? HELP - Get help using Pine
- C COMPOSE MESSAGE - Compose and send a message
- I FOLDER INDEX - View messages in current folder
- L FOLDER LIST - Select a folder to view
- A ADDRESS BOOK - Update address book
- S SETUP - Configure or update Pine
- Q QUIT - Exit the Pine program

Copyright 1989-1994. PINE is a trademark of the University of Washington.

? Help P PrevCmd R RelNotes
O OTHER CMDS L [ListFldrs] N NextCmd K KBLock

PINE is set up as a menu interface. You can select an option by using the cursor keys, the 'P' and 'N' commands, and the letter associated with each command. For example, typing 'I' brings up the following screen:

PINE 3.91 FOLDER INDEX

Folder: INBOX Message 1 of 23 NEW

```

N 1 Jan 30 Laura J. Mulcahy (1,555) Stolen with a screwdriver
+ N 2 Jan 30 Julie Joann Brain (1,246) Hi
N 3 Jan 30 Eric Mercer (728) RE: Stolen with a screwdriver
+ N 4 Jan 30 NEUS@mimi@magic.it (4,831) Commodore Trivia Edition #25 answers
N 5 Jan 30 Gordon Wilson (2,422) Re: Re[2]: Is there someone???
N 6 Jan 30 Dan Neuwirth (1,380) Re: Stolen with a screwdriver
N 7 Jan 30 Johnny H. Lee (EXC (2,282) RE: Stolen with a screwdriver
N 8 Jan 30 Kirkwood, Matt (2,525) RE: Roof repair advice...
+ N 9 Jan 30 M.W. Cottrell (1,451) banner for magazine
N 10 Jan 30 TonyJess@aol.com (1,421) Re: Roof repair advice...
N 11 Jan 30 Laura J. Mulcahy (1,591) Screwdriver, part II
+ N 12 Jan 30 aaron@yahoo.com (1,322) Yahoo Change
N 13 Jan 30 John Bolhuis (1,923) Re: Stolen Saturns
N 14 Jan 30 Johnny H. Lee (EXC (2,598) RE: Roof repair advice...
N 15 Jan 30 Machnik, Heather (1,844) repairing plastic panels
N 16 Jan 30 Ray Prill (2,938) Re: repairing plastic panels
+ N 17 Jan 30 Chris Stanford (1,230) Comments via Mosaic
N 18 Jan 30 Machnik, Heather (1,191) Screwdriver, part II
N 19 Jan 30 Kyle Koppenhoefer (2,045) Re: repairing plastic panels

```

? Help M Main Menu P PrevMsg - PrevPage D Delete R Reply
O OTHER CMDS V View Msg N NextMsg Spc NextPage U Undelete F Forward

Replying to a message is as simple as selecting the message with the selection methods outlined earlier, and typing 'R'.

New users can find their way easily in PINE, but more advanced users might prefer other shell email programs. For them, elm, mh, and mail are provided.

OK, we are out of time. For your homework, I would like an email message from each class member. Address the message to brain@mail.msen.com and make the Subject:

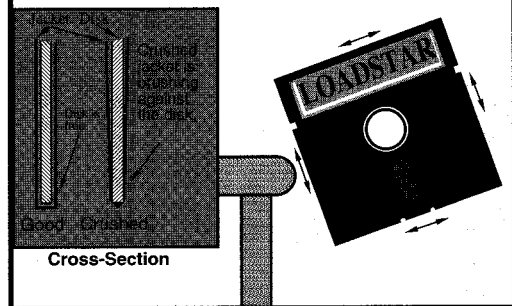
"LOADSTAR TEST MESSAGE".

Jim Brain

brain@mail.msen.com

Disk Trouble?

Every copy of LOADSTAR leaves J & F Publishing as a verified okay disk set. 3.5-inch disks are virtually trouble-free because they are difficult to damage, even when mishandled. Unfortunately some 5.25-inch disks may become crushed in the mail and give READ ERRORS because they have trouble spinning. Fixing these crushed disks is quite easy. Just run the areas indicated along a hard edge such as a table. This should uncrimp the disk.



THE UNDERGROUND

The 8-bit Commodore Mini-Mag!

The Underground is a bi-monthly publication with articles featuring reviews, PD/Shareware, BASIC and machine language tutorials, Geos, telecommunications, international stuff, news and tips. The Underground is a publication created by people who refuse to let their 8-bit computers gather dust, and is priced low enough for even the most frugal users. Try a sample issue to see if the Underground is worth your while. You'll be glad you did.

Single Issue \$2.00
Can/Mex \$2.25 • International \$3.75
One Year (six issues) \$11.00
Can/Mex 12.50 • International \$21.50
All prices in US Funds

Send check or Money Order (US funds) to:

THE UNDERGROUND

4574 Via Santa Maria
Santa Maria, CA 93455
USA

HD Series SCSI Hard Disk Drives

HD-40, 42 MB (Limited Supply)	\$249.00
HD-500, 500+ MB (Special Edition)	\$399.00
HD-1000, 1 GB (Special Edition)	\$499.00
HD-2000, 2 GB (Special Edition)	\$899.00

FD Series 3.5" Floppy Disk Drives

FD-2000 (800K and 1.6 MB)	\$179.95
FD Real-Time-Clock Option/Kit	\$20.00/\$29.00
Box of 10, High Density Disks (1.6MB)	\$14.95
Box of 10, Enhanced Density Disks (3.2 MB)	\$29.00

JiffyDOS

(Specify computer serial number and drive model)	
JiffyDOS C64/SX-64 System	\$49.95
JiffyDOS C-128/128-D System	\$59.95
Additional Drive ROM's	\$24.95

Miscellaneous Hardware

80-column Monitors (Refurb)	CALL
Aprotek 3-Way User Port Expander	\$30.00
Aprotek User Port Extension Cable	\$19.00
C-64/64-C Computers (Refurb, w/JD)	\$89.00
C-64, C-64C Power Supply (Repairable)	\$39.00
C-128 Power Supply (Repairable)	CALL
Cannon BJC-4100 Color Bubble Jet Printer	\$399.00
Commodore 1541 Disk Drives (Refurb, w/JD)	\$75.00
Commodore 1541-II Disk Drive (Refurb, w/JD)	\$89.00
Commodore 1571 Disk Drive (Refurb, w/JD)	\$99.00
Commodore 1802 40-column Monitor (Refurb)	\$149.00
CMD EX2-1 3-Port Cartridge Port Expander	\$34.95
CMD EX3 3-Port Cartridge Port Expander	\$29.95
CMD Gamepad/Joystick Controller	\$24.95
CMD GeoCable II Print Cable 6ft./15ft.	\$29.00/\$34.00
CMD SmartMouse (1351 Compatible Mouse)	\$49.95
CMD SmartTrack (1351 Compatible Trackball)	\$69.95
Inkwell Light Pen Model 170C	\$75.00
Monitor Cables	CALL
Mouse Pad	\$2.95
MW-350 Printer Interface (20K/8K Buffer)	\$49.00/\$60.00
Panasonic KXP-1150 9-pin Printer	\$169.00
Panasonic KXP-2023 24-pin Printer	\$229.00
Panasonic KXP-2135 24-pin Color Printer	\$319.00

Utilities

Big Blue Reader V4.10 (SOGWAP)	\$39.00
CMD Utilities	\$24.95
JiffyMON-64 (ML Monitor)	\$19.95
The Compression Kit '94 (Mad Man)	\$39.00

GEOS

Collette Utilities (Handy Geos Utilities)	\$19.95
Desk Pack Plus	\$29.00
Dweezils Greatest Hits (NewTools2, Stamp, Label128)	\$30.00
FONTPACK Plus	\$25.00
gateWay 64 or 128 (Specify Version)	\$29.95
geoBASIC	\$20.00
geoCalc 64/128	\$40.00/\$45.00
geoChart	\$29.00
geoFAX	\$39.95
geoFile 64/128	\$40.00/\$45.00
geoMakeBoot (Makes Bootable copies)	\$12.95
geoProgrammer	\$45.00
geoPublish	\$40.00
GEOS 64 v2.0	\$44.00
GEOS 128 v2.0	\$49.00
geoSHELL V2.2 (CLI for GEOS)	\$24.95
International FONTPACK	\$25.00
Perfect Print LQ for GEOS (Laser-like output)	\$49.95
RUN GEOS Companion	\$20.00
RUN GEOS Power Pak I or II (Specify)	\$20.00

Books

Anatomy of the 1541	\$12.00
C-64 Science & Engineering	\$12.00
C128 Computer Aided Design	\$12.00
C128 BASIC Training Guide	\$12.00
Commodore 64 Tricks and Tips	\$12.00
GEOS Programmers Reference Guide	\$35.00
Graphics Book for the C-64	\$12.00
Hitchhikers Guide to GEOS	\$35.00
Ideas for Use on Your C-64	\$12.00
Printer Book for the C64	\$12.00
Mapping the C64	\$12.00
Simple Internet	\$16.95
Superbase - The Book	\$15.00

Scanning & Video

Handyscanner	\$249.00
Pagefox	\$139.00
Video Digitizer	\$189.00
Videofox	\$89.00

RAMLink Power-Backed RAM Disk

RAMLink Base Model (2MB, No RAMCard)	\$149.00
RAMLink w/1 MB RAMCard (Special)	\$199.00
RAMLink w/4 MB RAMCard (Special)	\$299.00
RAMCard RTC Option/Kit	\$10.00/\$19.00
RAMLink Battery Back-up (Optional)	\$24.95
Parallel Cable (RAMLink to HD)	\$14.95

MIDI & Sound

Digimaster	\$34.95
SID Symphony Stereo Cartridge	\$44.95
Sonus 64 Sequencer, MIDI Interface, Cables	\$99.00
Sonus 128 Sequencer, MIDI Interface, Cables	\$99.00

Languages & Compilers

BASIC 64 Compiler (Abacus)	\$17.00
BASIC 128 Compiler (Abacus)	\$25.00
Blitzl 64 Compiler (Skyles)	\$30.00
Blitzl 128 Compiler (Skyles)	\$30.00
Buddy 64/128 Assembler	\$39.00
Cobol 64 (Abacus)	\$17.00
Fortran 64 (Abacus)	\$17.00
Pascal 64 (Abacus)	\$17.00
Power C 64 (Spinnaker)	\$14.00

Productivity

Cadpak 64 (Abacus)	\$22.00
Cadpak 128 (Abacus)	\$25.00
Chartpak 64 (Abacus)	\$17.00
Chartpak 128 (Abacus)	\$25.00
Data Manager 64 (Timeworks)	\$16.00
I Paint v1.5 (128, 80-col, 64K VDC) (Living Proof)	\$39.00
I Port v1.54 (128, 80-col, 64K VDC) (Living Proof)	\$29.00
Personal Portfolio Manager (Abacus)	\$16.00
Pocket Writer 1 (64) Digital Sol.	\$35.00
Pocket Writer 2 (64/128) (Digital Sol.)	\$65.00
Pocket Writer 3 (64 or 128) (Digital Sol.)	\$70.00
Pocket Planner 2 or Pocket Filer 2 (Digital Sol.)	\$35.00
RUN Productivity Pak I, II, or III (Specify)	\$15.00
RUN Super Starter Pak 1541 or 1581	\$20.00
RUN Works	\$20.00
SEC Check Register 128	\$29.00
SuperScript 64 (Precision)	\$15.00
SuperScript 128 (Precision)	\$20.00
Superbase 64 Version 3.01 (Precision)	\$35.00
Superbase 128 Version 3.01 (Precision)	\$35.00
SwiftCalc 64 (Timeworks)	\$16.00
Perfect Tax 64 (Free '95 Upgrade)	\$69.00
Perfect Tax 128 (Free '95 Upgrade)	\$79.00
TWS 64 w/Speller (Busy Bee)	\$29.00
TWS 128 w/Speller (Busy Bee)	\$39.00
TWS Modules (HD/RL/Illustrator)	each \$5.00

Telecommunications

Aprotek MiniModem C-24 (C= ready, 2400 baud)	\$69.00
Aprotek MiniModem C (C= ready, 1200 baud)	\$50.00
Aprotek Modem adapter (C= to Ext. PC Modem)	\$20.00
BOCA 2400 Baud Modem	\$69.00
BOCA 2400 w/SwiftLink and Cable	\$99.00
BOCA 14.4K bps FaxModem	\$129.00
BOCA 14.4K w/SwiftLink & Cable	\$159.00
BOCA V.34 28.8K bps FaxModem	\$199.00
BOCA V.34 w/SwiftLink & Cable	\$229.00
Dialogue 128	\$29.00
SwiftLink RS-232 Cartridge (Up to 38.4K baud)	\$39.95
SwiftLink Modem Cable (DB9-DB25)	\$9.95

Games

Atomino	\$17.00
Ballistix	\$13.00
Blood Money	\$15.00
Chompl	\$10.00
Day in the Life of Prehistoric Man	\$19.00
Escape Route	\$19.00
Heavenbound	\$19.95
Island of the Dragon	\$19.00
Laser Squad	\$13.00
Lazer Duel	\$14.95
Lions of the Universe	\$19.95
Mainframe	\$13.00
Menace	\$15.00
Navy Seal	\$10.00
Rings of Medusa	\$16.00
RUN C128 Funpak	\$10.00
RUN C64 Gamepak	\$10.00
Skate or Die	\$13.00
Sisterman	\$14.95
The Amazing Spider-Man	\$15.00
The President is Missing!	\$10.00
The Three Stooges	\$10.00
Tie Break Tennis	\$16.00
Total Eclipse	\$10.00
Ultima V	\$17.00
Walkerz	\$19.00
Wings of Circe	\$19.00
Wizardry 5: Heart of the Maelstrom	\$24.00



JiffyDOS

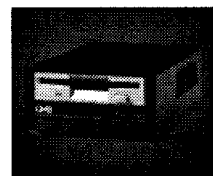
Increase Speed Up to 1500% while retaining 100% compatibility

- Speeds up Loading, Saving, Verifying, Formatting and Reading/ Writing of Program, Sequential, User and Relative files unlike cartridges which only speed up Loading and Saving of PRG files
- Built-in DOS Wedge plus 17 additional features including file copier, text dump, printer toggle, and redefinable function keys make using your computer easier and more convenient
- ROM upgrade installs easily into most computers and disk drives. Supports C-64, 64C, SX-64, C-128, 128-D, 1541, 1541C, 1541-II, 1571, 1581 and more.

FD Series

1.6 MB 3.5" Floppy Drives

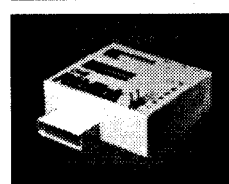
The FD-2000 disk drives utilize today's latest 3.5 inch disk drive technology. FD-2000's support 800K (1581 style) and 1.6 MB (High Density) formats. Fast and reliable, they support 1541, 1571 and 1581 style partitions. Native Mode partitioning and can actually read and write 1581 disks. FD's feature built-in JiffyDOS, SWAP button and optional Real Time Clock. High capacity, speed and compatibility make the FD right for every application, including GEOS.



RAMLink

Power-Backed Expandable REU

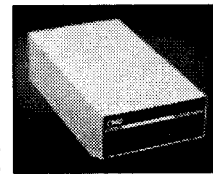
The fastest possible form of storage, RAMLink provides instant access to files and retains data while your computer is turned off. Easy to use and expandable up to 16 MB, RAMLink utilizes the same powerful operating system found in CMD Hard Drives. Unlike Commodore REU's which are compatible with less than 1% of commercial software, RAMLink supports more than 80% of the commercial titles. RAMLink also offers built-in JiffyDOS Kernel, SWAP feature, reset button, enable/disable switch, pass-thru port and RAM port for REU/GEORAM use. RAMLink offers maximum speed, expandability and compatibility with all types of software and hardware including GEOS.



HD Series

SCSI Hard Drive for the C-64/128

HD Series Hard Drives are available in capacities up to 1 GB, are fully partitionable, and can emulate 1541, 1571, & 1581 disks while Native partitions utilize MS-DOS-style subdirectories. HD's connect easily to the serial bus or parallel via RAMLink. Includes built-in JiffyDOS, SWAP feature and RTC. HD's offer superior compatibility with most commercial software including BBS, Productivity and GEOS. And with new pricing, HD Series drives offer the lowest cost per megabyte of any C64/128 storage device.



Shipping and Handling Charges

Use the chart below to match your order subtotal with your shipping zone and method

Order Subtotal	Continental United States				
	UPS Ground	2nd Day	Next Day	AK HI, PR	Canada
\$0.01 - \$19.99	\$3.00	\$8.00	\$17.00	\$12.00	\$5.00
\$20.00-\$29.99	\$5.00	\$9.00	\$18.00	\$14.00	\$7.00
\$30.00-\$59.99	\$6.00	\$10.00	\$20.00	\$15.00	\$9.00
\$60.00-\$149.99	\$8.00	\$12.00	\$23.00	\$19.00	\$12.00
\$150.00-\$299.99	\$10.00	\$14.00	\$27.00	\$21.00	\$20.00
\$300.00-\$799.99	\$15.00	\$20.00	\$34.00	\$27.00	\$25.00
\$800.00 +	\$20.00	\$25.00	\$40.00	\$32.00	\$35.00

UPS C.O.D. add \$5.00 (U.S./PR only)

Policies: CMD accepts: Major Credit Cards, Money Orders, COD and Personal Checks. Personal Checks are held 3 weeks. Customer is responsible for shipping charges on refusals. Most items are stock, contact CMD for firm delivery. Returns for merchandise credit only within 30 days with prior authorization. Credits are less shipping, handling, taxes, customs, duties, and a 10% restocking charge. No refunds or credits on opened software. All prices and specifications are subject to change without notice.

WE ACCEPT ALL MAJOR CREDIT CARDS

CMD Creative Micro Designs, Inc.

P.O. Box 646
East Longmeadow, MA 01028

Info: (413) 525-0023
Fax: (413) 525-0147